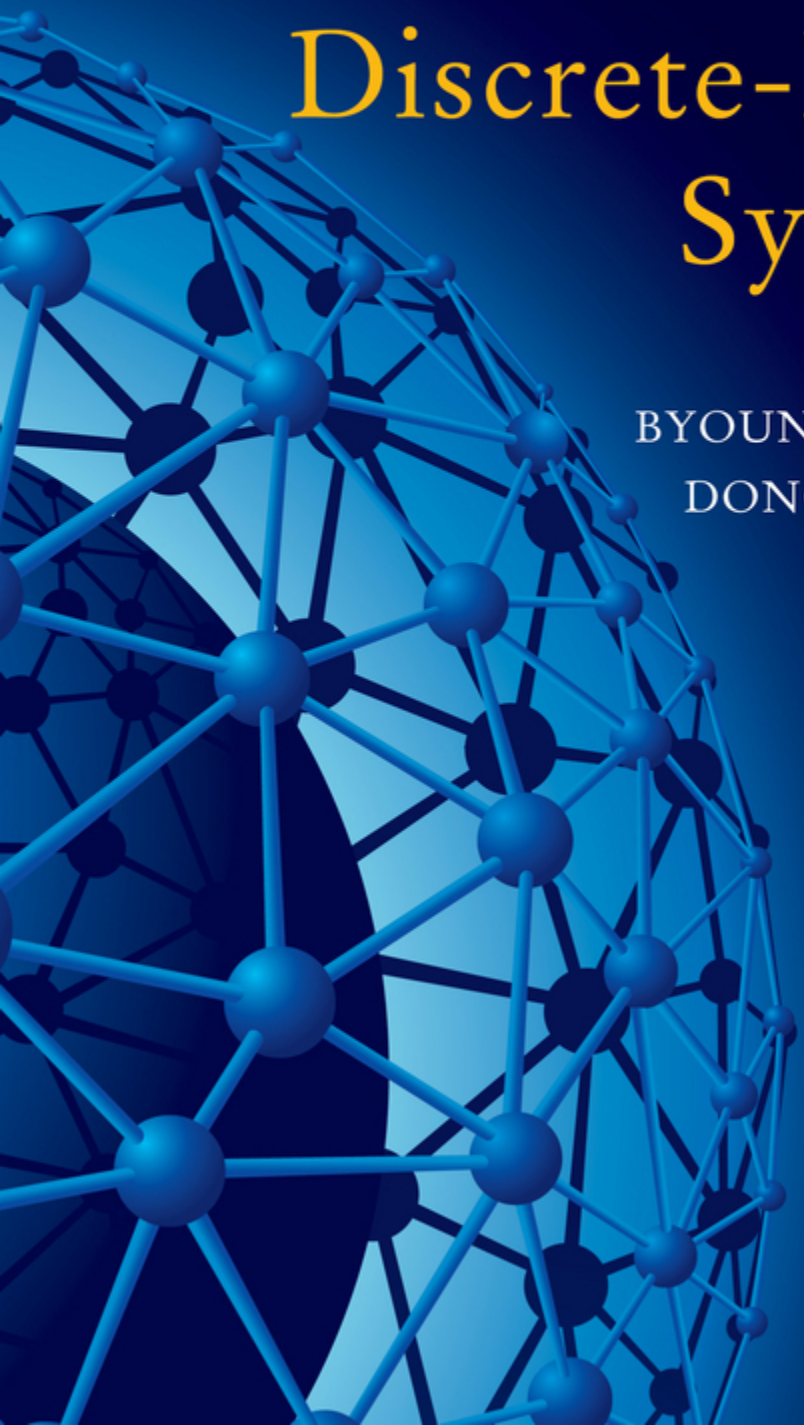
Modeling *and* Simulation *of* Discrete-Event Systems

BYOUNG KYU CHOI
DONGHUN KANG



WILEY

MODELING AND SIMULATION OF DISCRETE-EVENT SYSTEMS

MODELING AND SIMULATION OF DISCRETE-EVENT SYSTEMS

Byoung Kyu Choi
Department of Industrial and Systems Engineering, KAIST, South Korea
Department of Computer Science, KAU, Saudi Arabia

Donghun Kang
Department of Industrial and Systems Engineering, KAIST, South Korea

WILEY

Cover image: 08-17-09 © Mark Divers (iStock photo ID: 10295380)

Copyright © 2013 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey
Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the Web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our website at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:
Choi, Byoung Kyu, 1949–
Modeling and simulation of discrete-event systems / Byoung Kyu Choi, Donghun Kang.
pages cm
Includes index.
ISBN 978-1-118-38699-6 (cloth)
1. Discrete-time systems–Simulation methods. I. Kang, Donghun, 1981– II. Title.
T57.62.C377 2013
003'.83–dc23
2013013970

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

CONTENTS

PREFACE xvii

ABBREVIATIONS xix

PART I BASICS OF SYSTEM MODELING AND SIMULATION 1

1. Overview of Computer Simulation 3

1.1 Introduction 3
1.2 What Is a System? 4
1.2.1 Definitions of Systems 4
1.2.2 Three Types of Systems 4
1.2.3 System Boundaries and Hierarchical Structure 5
1.3 What Is Computer Simulation? 6
1.3.1 What Is Simulation? 6
1.3.2 Why Simulate? 7
1.3.3 Types of Computer Simulation 8
1.4 What Is Discrete-Event Simulation? 9
1.4.1 Description of System Dynamics 10
1.4.2 Simulation Model Trajectory 10
1.4.3 Collecting Statistics from the Model Trajectory 11
1.5 What Is Continuous Simulation? 11
1.5.1 Manual Simulation of the Newtonian Cooling Model 12
1.5.2 Simulation of the Newtonian Cooling Model Using a Simulator 12
1.6 What Is Monte Carlo Simulation? 12
1.6.1 Numerical Integration via Monte Carlo Simulation 12
1.6.2 Risk Analysis via Monte Carlo Simulation 14
1.7 What Are Simulation Experimentation and Optimization? 15
1.8 Review Questions 16

2. Basics of Discrete-Event System Modeling and Simulation 17

2.1 Introduction 17
2.2 How Is a Discrete-Event Simulation Carried Out? 17
2.2.1 Event Routines 18
2.2.2 Simulation Model Trajectory 19
2.2.3 Manual Simulation Execution 19
2.2.4 Flow Chart of Manual Simulation Procedure 22
2.3 Framework of Discrete-Event System Modeling 23
2.3.1 What Are Modeling Components and Reference Model? 23
2.3.2 What Is a Discrete-Event System (DES) Modeling Formalism? 24
2.3.3 What Is a Formal Model and How Is It Specified? 26
2.3.4 Integrated Framework of DES Modeling 29
2.4 Illustrative Examples of DES Modeling and Simulation 32
2.4.1 How to Build and Simulate an Event Graph Model of a DES 33
2.4.2 How to Build and Simulate an ACD Model of a DES 35
2.4.3 How to Build and Simulate a State Graph Model of a DES 37
2.5 Application Frameworks for Discrete-Event System Modeling and Simulation 38
2.5.1 How Is the M&S Life Cycle Managed? 38
2.5.2 Framework for Factory Life-Cycle Support 39
2.6 What to Cover in a Simulation Class 40
2.6.1 Event-Based M&S and Event-Graph Simulation with SIGMA® 41
2.6.2 Activity-Based M&S and Hands-On Modeling Practice with Arena® 41
2.6.3 State-Based M&S 41
2.7 Review Questions 42

PART II FUNDAMENTALS OF DISCRETE-EVENT SYSTEM MODELING AND SIMULATION 43

3. Input Modeling for Simulation 45

3.1 Introduction 45
3.2 Empirical Input Modeling 46
3.2.1 Nonparametric Modeling 46

3.2.2 Empirical Modeling of Individual Data 46
3.2.3 Empirical Modeling of Grouped Data 47
3.3 Overview of Theoretical Distribution Fitting 48
3.3.1 Data Independence Checking 48
3.3.2 Distribution Function Selection 49
3.3.3 Parameter Estimation 49
3.3.4 Goodness-of-Fit Test 49
3.3.5 Overview of Random Variate Generation 49
3.4 Theoretical Modeling of Arrival Processes 50
3.4.1 Theoretical Basis for Arrival Process Modeling 50
3.4.2 Generation of Inter-Arrival Times for a Constant Arrival Rate 51
3.4.3 Generation of Inter-Arrival Times for Varying Arrival Rates 52
3.5 Theoretical Modeling of Service Times 53
3.5.1 Generation of Service Time in the Absence of Data 53
3.5.2 Generation of Service Times from Collected Data 55
3.6 Input Modeling for Special Applications 57
3.6.1 Interfailure Time Modeling 57
3.6.2 Inspection Process Modeling 58
3.6.3 Batch Size Modeling 59
3.7 Review Questions 59
Appendix 3A: Parameter Estimation 60
3A.1 Exponential Distribution 60
3A.2 Erlang Distribution 60
3A.3 Beta Distribution 61
3A.4 Weibull Distribution 62
3A.5 Normal and Lognormal Distributions 64
Appendix 3B: Random Variate Generation 64
3B.1 Exponential Random Variate 64
3B.2 Erlang Random Variate 65
3B.3 Beta Random Variate 65
3B.4 Weibull Random Variate 66
3B.5 Normal and Lognormal Random Variates 67
3B.6 Triangular Random Variate 67

4. Introduction to Event-Based Modeling and Simulation 69

4.1 Introduction 69
4.2 Modeling and Simulation of a Single Server System 70

4.2.1 Reference Modeling 70
4.2.2 Formal Modeling 71
4.2.3 Model Execution 72
4.3 Execution Rules and Specifications of Event Graph Models 72
4.3.1 Event Graph Execution Rules 72
4.3.2 Tabular Specification of Event Graph Models 73
4.3.3 Algebraic Specifications of an Event Graph Model 75
4.4 Event Graph Modeling Templates 75
4.4.1 Single Queue Models 76
4.4.2 Tandem Line Models 80
4.5 Event Graph Modeling Examples 82
4.5.1 Flexible Multi-Server System with Fluctuating Arrival Rates 82
4.5.2 Car Repair Shop 82
4.5.3 Project Management Modeling 84
4.5.4 Conveyor-Driven Serial Line 85
4.5.5 Inline-Type Manufacturing Cell Modeling 86
4.6 Execution of Event Graph Models with SIGMA 91
4.6.1 Simulation of a Single Server System with SIGMA 92
4.6.2 Simulation of a Conveyor-Driven Serial Line with SIGMA 95
4.7 Developing Your Own Event Graph Simulator 99
4.7.1 Functions for Handling Events and Managing Queues 99
4.7.2 Functions for Generating Random Variates 101
4.7.3 Event Routines 101
4.7.4 Next Event Methodology of Simulation Execution 102
4.7.5 Single Server System Simulator 103
4.8 Review Questions 106

5. Parameterized Event Graph Modeling and Simulation 107

5.1 Introduction 107
5.2 Parameterized Event Graph Examples 108
5.2.1 Introducing Index Variables to a Repeating Event-Vertex Pattern 108
5.2.2 Passing Attribute Values of Each Entity along Event Vertices 109

5.3 Execution Rules and Specifications of the Parameterized Event Graph 110
5.3.1 Execution Rules of the PEG Model 110
5.3.2 Tabular Specifications of the PEG Model 110
5.3.3 Algebraic Specifications of the PEG Model 111
5.4 Parameterized Event Graph Modeling of Tandem Lines 112
5.4.1 PEG Modeling of an Unlimited Buffer Tandem Line 112
5.4.2 PEG Modeling of a Limited Buffer Tandem Line 113
5.4.3 PEG Modeling of a Conveyor-Driven Serial Line 114
5.5 Parameterized Event Graph Modeling of Job Shops 115
5.5.1 PEG Modeling of a Simple Job Shop without Transport 115
5.5.2 PEG Modeling of a Job Shop with Transport and Setup Times 117
5.5.3 PEG Modeling of an Inline Job Shop 118
5.5.4 PEG Modeling of a Mixed Job Shop 121
5.6 Execution of Parameterized Event Graph Models Using SIGMA 122
5.6.1 Collecting Sojourn Time Statistics Using SIGMA Functions 123
5.6.2 Simulating a Simple Service Shop with SIGMA 126
5.6.3 Simulation of a Three-Stage Tandem Line Using SIGMA 128
5.6.4 Simulation of the Simple Job Shop with SIGMA 131
5.7 Developing Your Own Parameterized Event Graph Simulator 137
5.7.1 Tandem Line PEG Simulator 137
5.7.2 Simple Job Shop PEG Simulator 140
5.8 Review Questions 142

6. Introduction to Activity-Based Modeling and Simulation 143

6.1 Introduction 143
6.2 Definitions and Specifications of an Activity Cycle Diagram 145
6.2.1 Definitions of an ACD 146
6.2.2 Execution Rules and Tabular Specifications of an ACD 147
6.2.3 Algebraic Specifications of an ACD 148

6.3 Activity Cycle Diagram Modeling Templates 150
6.3.1 ACD Template for Flexible Multi-Server System Modeling 151
6.3.2 ACD Template for Limited Buffer Tandem Line Modeling 152
6.3.3 ACD Template for Nonstationary Arrival Process 153
6.3.4 ACD Template for Batched Service Modeling 153
6.3.5 ACD Template for Joining Operation Modeling 154
6.3.6 ACD Template for Probabilistic Branching Modeling 154
6.3.7 ACD Template for Resource Failure Modeling 155
6.4 Activity-Based Modeling Examples 156
6.4.1 Activity-Based Modeling of a Worker-Operated Tandem Line 156
6.4.2 Activity-Based Modeling of an Inspection-Repair Line 157
6.4.3 Activity-Based Modeling of a Restaurant 158
6.4.4 Activity-Based Modeling of a Simple Service Station 159
6.4.5 Activity-Based Modeling of a Car Repair Shop 160
6.4.6 Activity-Based Modeling of a Project Management System 161
6.4.7 Activity-Based Modeling of a Conveyor-Driven Serial Line 161
6.5 Parameterized Activity Cycle Diagram and Its Application 163
6.5.1 Definition and Specifications of Parameterized ACD 163
6.5.2 Rules for Executing the P-ACD Model 164
6.5.3 P-ACD Modeling of Tandem Lines 165
6.5.4 P-ACD Modeling of Job Shops 168
6.6 Execution of Activity Cycle Diagram Models with a Formal Simulator ACE® 171
6.6.1 Simulation of Single Server Model with ACE 171
6.6.2 Simulation of Probabilistic Branching Model with ACE 175
6.6.3 Simulation of Resource Failure Model with ACE 176
6.6.4 Simulation of Simple Service Station Model with ACE 180
6.7 Review Questions 183

7. Simulation of ACD Models Using Arena® 184

7.1 Introduction 184
7.2 Arena Basics 185
7.2.1 Arena Modeling Environment 186
7.2.2 Building a Flowchart Model of a Process-Inspect Line 187
7.2.3 Completing a Static Model of a Process-Inspect Line 191
7.2.4 Arena Simulation and Output Reports 192
7.2.5 Arena Modules 194
7.3 Activity Cycle Diagram-to-Arena Conversion Templates 197
7.3.1 Template for Fixed Multi-Server Modeling 198
7.3.2 Template for Flexible Multi-Server Modeling 201
7.3.3 Template for Balking (Conditional Branching) Modeling 202
7.3.4 Template for Limited Buffer Tandem Line Modeling 204
7.3.5 Template for Nonstationary Arrival Process Modeling 205
7.3.6 Template for Joining Operation Modeling 206
7.3.7 Template for Inspection (Probabilistic Branching) Modeling 207
7.3.8 Template for Resource Failure Modeling 208
7.4 Activity Cycle Diagram-Based Arena Modeling Examples 209
7.4.1 ACD-Based Arena Modeling of a Worker-Operated Tandem Line 210
7.4.2 ACD-Based Arena Modeling of Restaurant 211
7.4.3 ACD-Based Arena Modeling of a Simple Service Station 213
7.4.4 ACD-Based Arena Modeling of a Project Management System 214
7.4.5 ACD-Based Arena Modeling of a Job Shop 216
7.4.6 ACD-Based Arena Modeling of a Conveyor-Driven Serial Line 219
7.5 Review Questions 223

8. Output Analysis and Optimization 224

8.1 Introduction 224
8.2 Framework of Simulation Output Analyses 225
8.2.1 Verification and Calibration 225

8.2.2 Simulation Experimentation 226
8.2.3 Communication and Presentation 227
8.3 Qualitative Output Analyses 228
8.4 Statistical Output Analyses 230
8.4.1 Statistical Output Analyses for Terminating Simulations 230
8.4.2 Statistical Output Analyses for Nonterminating Simulations 231
8.4.3 Statistical Output Analyses for Comparing Alternative Systems 233
8.5 Linear Regression Modeling for Output Analyses 234
8.5.1 Linear Regression Models 234
8.5.2 Regression Parameter Estimation 235
8.5.3 Test for Significance of Regression 236
8.5.4 Linear Regression Modeling Example 238
8.5.5 Regression Model Fitting for Qualitative Variables 240
8.6 Response Surface Methodology for Simulation Optimization 241
8.6.1 Overview of RSM for Process Optimization 241
8.6.2 Searching for Optimum Regions with the Steepest Ascent 241
8.6.3 Second-Order Model Fitting for Optimization 245
8.7 Review Questions 247
Appendix 8A: Student's *t*-Distribution 248
8A.1 Definition 248
8A.2 Derivation of the *t*-Statistic 248
8A.3 Table of Critical *t*-Values with Degrees of Freedom (df) 248
Appendix 8B: Student's *t*-Tests 249
8B.1 One Sample *t*-Test 249
8B.2 Unpaired Two Sample *t*-Test 250

PART III ADVANCES IN DISCRETE-EVENT SYSTEM MODELING AND SIMULATION 253

9. State-Based Modeling and Simulation 255

9.1 Introduction 255
9.2 Finite State Machine 256
9.2.1 Existing Definitions of Finite State Machines 256

9.2.2 Finite State Machine Models 257
9.2.3 Finite State Machine Modeling of Buffer Storage and Single Server Systems 258
9.2.4 Execution of Finite State Machine Models 259
9.3 Timed Automata 261
9.3.1 Language and Automata 261
9.3.2 Timed Automata 262
9.3.3 Timed Automata with Guards 263
9.3.4 Networks of Timed Automata 266
9.4 State Graphs 267
9.4.1 State Variables and Macro States 267
9.4.2 Timers and System Variables 268
9.4.3 Conventions for Building State Graphs and State Transition Tables 269
9.5 System Modeling with State Graph 271
9.5.1 State Graph Modeling of Dining Philosophers 271
9.5.2 State Graph Modeling of a Table Tennis Game 272
9.5.3 State Graph Modeling of a Tandem Line 275
9.5.4 State Graph Modeling of a Conveyor-Driven Serial Line 275
9.5.5 State Graph Modeling of Traffic Intersection Systems 279
9.6 Simulation of Composite State Graph Models 283
9.6.1 Framework of a State Graph Simulator 283
9.6.2 Synchronization Manager 284
9.6.3 Atomic Simulators 287
9.6.4 Table Tennis Game Simulator 290
9.6.5 State Graph Simulator for Reactive Systems 293
9.6.6 SGS® 295
Appendix 9A: DEVS 295
9A.1 Definitions of DEVS 295
9A.2 DEVS Simulators 297

10. Advanced Topics in Activity-Based Modeling and Simulation 299

10.1 Introduction 299
10.2 Developing Your Own Activity Cycle Diagram Simulators 300
10.2.1 Tocher's Three-Phase Process 300
10.2.2 Activity Scanning Algorithm 302
10.2.3 ACD Simulator 304

10.2.4 P-ACD Simulator 306
10.2.5 Collecting Statistics 309
10.3 Modeling with Canceling Arc 310
10.3.1 ACD Model of Single Server System with Reneging 311
10.3.2 ACD Model of Resource Failure 312
10.3.3 ACD Model of Time-Constrained Processing 313
10.3.4 Execution of Canceling Arc 313
10.4 Cycle Time Analysis of Work Cells via an Activity Cycle Diagram 313
10.4.1 Cycle Time Analysis of Single-Armed Robot Work Cell 314
10.4.2 Cycle Time Analysis of Single Hoist Plating Line 316
10.4.3 Cycle Time Analysis of Dual-Armed Robot Cluster Tool 319
10.5 Activity Cycle Diagram Modeling of a Flexible Manufacturing System 322
10.5.1 ACD Modeling of Job Flows in FMS 323
10.5.2 P-ACD Modeling of Job Routing in FMS 323
10.5.3 P-ACD Modeling of AGV Dispatching Rules in FMS 325
10.5.4 P-ACD Modeling of Refixture Operation and Heterogeneous FMS 327
10.6 Formal Model Conversion 329
10.6.1 Conversion of ACD Models to Event Graph (EG) Models 329
10.6.2 Conversion of ACD Models to State Graph (SG) Models 330
10.6.3 Examples of Formal Model Conversion 331
Appendix 10A: Petri Nets 334
10A.1 Definitions of Petri Nets 334
10A.2 Petri-Net State and Execution 335
10A.3 Extended Petri Nets and the ACD 336
10A.4 Restricted Petri Nets 337
10A.5 Modeling with Petri Nets 337

11. Advanced Event Graph Modeling for Integrated Fab Simulation 338

11.1 Introduction 338
11.2 Flat Panel Display Fabrication System 339
11.2.1 Overview of FPD Fab 339
11.2.2 FPD Processing Equipment 340
11.2.3 Material Handling System 342

11.3 Production Simulation of a Flat Panel Display Fab 343
11.3.1 Modeling of Uni-Inline Job Shop 343
11.3.2 Modeling of Oven Type Job Shop 345
11.3.3 Modeling of Heterogeneous Job Shop 346
11.3.4 Object-Oriented Event Graph Simulator for Production Simulation 346
11.4 Integrated Simulation of a Flat Panel Display Fab 350
11.4.1 Modeling of Job Shop for Integrated Simulation 350
11.4.2 Modeling of Conveyor Operation 353
11.4.3 Modeling of the Interface between Conveyor and Inline Stocker 355
11.4.4 Modeling of the Interface between Uni-inline Cells and Inline Stocker 357
11.4.5 Modeling of the Interface between Oven and Inline Stocker 358
11.4.6 Modeling of Inline Stocker Operation 358
11.4.7 Integrated Fab Simulator 361
11.5 Automated Material Handling Systems-Embedded Integrated Simulation of Flat Panel Display Fab 362
11.5.1 Concept of AMHS-Embedded Fab Simulation 363
11.5.2 Framework of AMHS-Embedded Fab Simulation System 364
11.5.3 Simulator for AMHS-Embedded Integrated Fab Simulation 366
11.5.4 IFS® 368

12. Concepts and Applications of Parallel Simulation 371

12.1 Introduction 371
12.2 Parallel Simulation of Workflow Management System 372
12.2.1 Enactment Service Mechanism of WfMS 372
12.2.2 Framework of Parallel Simulation of WfMS 373
12.2.3 State Graph Modeling of an Enactment Server and Sync Manager 375
12.2.4 State Graph Modeling of Participant Simulators 377
12.2.5 Implementation of a Workflow Simulator 377
12.3 Overview of High-Level Architecture/Run-Time Infrastructure 378
12.3.1 Basics of HLA/RTI 379
12.3.2 HLA Federation Architecture 381
12.3.3 Overview of Federation Execution 382

12.4 Implementation of a Parallel Simulation with High-Level Architecture/Run-Time Infrastructure 383
12.4.1 The Sushi Restaurant Federation 383
12.4.2 Preparation of an FED File 384
12.4.3 Preparation of the Federate Code (of the Production Federate) 386
12.4.4 Executing the Restaurant Federation 391

REFERENCES **395**

INDEX **400**

Online Supplements

Numerous supplemental materials including software downloads are provided on the official website of the book at http://VMS-technology.com/book. The supplemental materials are grouped into (1) M&S practices with commercial simulators, (2) developing your own dedicated simulators, and (3) integrated simulation of electronics Fabs. The commercials simulators covered are an event-based simulator SIGMA®, an activity-based simulator ACE®, a state-based simulator SGS®, and an entity-based simulator Arena®.

PREFACE

This book provides comprehensive, in-depth coverage of modeling and simulation (M&S) of discrete-event systems (DESs). Here, the term *M&S* refers to computer simulation, with an emphasis on modeling real-life DESs and executing the models. The current state-of-the-art in DES M&S is a result of the breakthroughs in the following areas: (1) activity-based modeling formalism pioneered by K.D. Tocher in late 1950s; (2) the advent of process-oriented simulation languages, such as GPSS and SLAM, in the early 1970s; (3) state-based modeling formalism, or DEVS, founded by Bernard Zeigler in the mid-1970s; and (4) event-based modeling formalism as matured by Lee Schruben since the early 1980s.

There exists at least one classic textbook in each area—a textbook on activity-based modeling by Carrie, a few books on state-based (DEVS) modeling by Zeigler, a textbook on event-based modeling by Schruben, and a few books on process-oriented languages such as Arena® and ProModel®. In addition, there are quite a few books focusing on statistical notions of computer simulation. The researchers in each area advocate their own views as central to DES M&S. Only a couple of books (e.g., Fishwick) propose an integrated model engineering framework.

This book presents an integrated M&S framework covering all four DES M&S breakthrough areas. It is a product of 30 years of teaching at KAIST, as well as sponsored research and development projects at the authors' lab at KAIST, VMS (virtual manufacturing system) Lab, which has been a government-endowed National Research Lab since 1999. In particular, the practice-oriented theme of this book is a result of the authors' decade-long experience in developing simulation-based scheduling (SBS) solutions for Samsung Electronics and other companies in Korea. Virtually all the Samsung's semiconductor fabrication plants (Fabs) and flat panel display (FPD) Fabs are run utilizing solutions originated by the authors' lab, and upgraded and supported by a spin-off venture company.

This book is divided into three parts: Part I, *Basics of System Modeling and Simulation*; Part II, *Fundamentals of Discrete-Event System Modeling and Simulation*; and Part III, *Advances in Discrete-Event System Modeling and Simulation*. Parts I and II are designed as a primary textbook for an undergraduate level M&S course in Industrial Engineering, Computer Science, and Management Science. With Part III, it is designed as a graduate-level course. This book comprehensively covers the state-of-the art modeling formalisms

and execution algorithms in DES M&S thereby serving as a main reference for M&S researchers in academia. This book provides an easy-to-understand guide for simulation practitioners in industry using off-the-shelf simulators such as SIGMA® and Arena®. Finally, this book reveals a number of "secrets" for developing your own simulators: event graph simulator, ACD simulator, state graph simulator, and integrated Fab simulator—making it a valuable resource for M&S solution developers.

The book is largely self-contained, and few prerequisites are needed for understanding its main contents. However, some prior knowledge will help readers understand specific sections:

(1) Basic knowledge of statistics and probability (Chapter 3, Input Modeling);
(2) Basic knowledge of linear algebra (Chapter 8, Output Analysis and Optimization)
(3) Experience with computer programming (Sections on developing your own simulators, e.g. Sections 5.7 and 10.2).

Perhaps the most critical prerequisite for mastering this book is enthusiasm and commitment toward M&S. This book is about the art of M&S, and like other art forms, can only be mastered through persistent practice.

The authors wish to express their special thanks to Prof. K.H. Han of Gyeongsang National University for using part of this book in his class and providing valuable comments that led to its improvement; to Prof. I.K. Moon of Seoul National University and Prof. S.C. Park of Ajou University for their input during the early stage of writing this book; and to Prof. Lee Schruben of Berkeley for his encouragement and support. For developing sample models and exercise problems, and for executing "prototype" simulation models appearing in the book, we would like to thank our graduate students in the VMS Lab at KAIST, especially H.S. Kim, T.J. Choi, and E.H. Song.

Finally, Byoung Choi thanks his wife, Yong, and his son and best friend Samuel, for their support and encouragement. Donghun Kang thanks his parents for their loving care and support.

Byoung Kyu Choi
Donghun Kang

Daejeon, Korea, June 2013

ABBREVIATIONS

ACD	activity cycle diagram
AGV	automated guided vehicle
AMHS	automated material handling system
AON	activity-on-node
AQL	average queue length
AST	average sojourn time
ATM	automatic teller machine
ATT	activity transition table
AWT	average waiting time
BTO	bound-to-occur
CAL	candidate activity list
DES	discrete-event system
EFD	entity flow diagram
EG	event graph
EO	event object
EOS	end of simulation
Fab	Fabrication line (or plant)
FED	federation execution data
FEL	future event list
FIFO	first-in-first-out
FMS	flexible manufacturing system
FOM	federation object model
FPD	flat panel display
FSA	finite state automata
FSM	finite state machine
HLA	high level architecture
IFS	integrated factory simulator
LCD	liquid crystal display
LEL	local event list
LP	logical process
M&S	modeling and simulation
MCS	material control system
MDP	message delivery packet
MEL	move-type event list
MLE	maximum likelihood estimator
MS	mean square
MSR	message send request
OOEG	object-oriented event graph
P-ACD	parameterized ACD
PDM	process definition model
PEG	parameterized event graph
RSM	response surface methodology
RTD	real-time dispatcher
RTI	run time infrastructure
SG	state graph
SOM	simulation object model
SS	sum of square
TAG	time advance grant
TAR	time advance request
VE	virtual environment
WfMS	workflow management system

PART I

BASICS OF SYSTEM MODELING AND SIMULATION

We think by "constructing mental models and then simulating them in order to draw conclusions or make decisions." Thus, modeling and simulation (M&S) constitutes the central part of our thinking process. "I think, therefore I am" is a philosophical statement used by the French philosopher Descartes, which became a foundational element of Western philosophy. Therefore, if we combine the philosophical notion of thinking with the engineering definition of M&S, we may say that "we are engineers and scientists because we can model systems and simulate them." Furthermore, if our brain is not powerful enough to simulate a given complex system, we rely on computers to perform a computer simulation.

A dictionary definition of *simulation* is the technique of imitating the behavior of some situation by means of an analogous situation or apparatus to gain information more conveniently or to train personnel, while an academic definition of *computer simulation* is the discipline of designing a model of a system, simulating the model on a digital computer, and analyzing the execution output. In recent years, the term *modeling and simulation* (M&S) seems to be preferred to the term for computer simulation, with an emphasis on modeling. Part I of this book has two chapters, and it aims to provide the readers with a basic but comprehensive treatment of computer simulation.

Chapter 1, "Overview of Computer Simulation," will provide answers to the following basic questions in computer simulation:

1. What is a system?
2. What is computer simulation?
3. What is discrete-event simulation?
4. What is continuous simulation?
5. What is Monte Carlo simulation?
6. What are simulation experimentation and optimization?

Modeling and Simulation of Discrete-Event Systems, First Edition. Byoung Kyu Choi and Donghun Kang.
© 2013 John Wiley & Sons, Inc. Published 2013 by John Wiley & Sons, Inc.

Chapter 2, "Basics of Discrete-Event System Modeling & Simulation," aims to provide answers to the following basic questions in discrete-event system (DES) M&S:

1. How is a discrete-event simulation carried out?
2. What are modeling components and a reference model?
3. What is a discrete-event system modeling formalism?
4. What is a formal model and how it is specified?
5. What is the integrated framework of discrete-event system modeling?
6. How do we build and simulate an event graph or activity cycle diagram (ACD) model of a DES?
7. How is the M&S life cycle managed?

CHAPTER 1

Overview of Computer Simulation

The wise man is one who knows what he does not know.

—Tao Te Ching

1.1 INTRODUCTION

Richmond [2003] defines thinking as "constructing mental models and then simulating them in order to draw conclusions or make decisions." Namely, he defines thinking as mental simulation. When the situation is too complex to be analyzed by mental simulation alone, we rely on computer simulation. According to Schruben [2012], simulation models provide unlimited virtual power: "If you can think of something, you can simulate it. Experimenting in a simulated world, you can change anything, in any way, at any time—even change time itself."

Fishwick [1995] defines *computer simulation* as the discipline of designing a model of a system, simulating the model on a digital computer, and analyzing the execution output. In the military, where computer simulation is extensively used in training personnel (e.g., war game simulation) and acquiring weapon systems (e.g., simulation-based acquisition), the term *modeling and simulation* (M&S) is used in place of *computer simulation*. In this book, these two terms are used interchangeably.

The purpose of this chapter is to provide the reader with a basic understanding of computer simulation. After studying this chapter, you should be able to answer the following questions:

1. What are the common characteristics that lead to a conceptual definition of system?
2. What are the three types of systems?
3. What are the three subsystems in a feedback control system?
4. What are the three types of virtual environment simulation?

Modeling and Simulation of Discrete-Event Systems, First Edition. Byoung Kyu Choi and Donghun Kang.
© 2013 John Wiley & Sons, Inc. Published 2013 by John Wiley & Sons, Inc.

5. What are the three types of computer simulation?
6. What is the simulation model trajectory of a discrete-event system?
7. What is Monte Carlo simulation?
8. What is sensitivity analysis in simulation experimentation?

This chapter is organized as follows: Definitions and structures of systems are given in Section 1.2. Section 1.3 provides definitions and applications of simulation. The subsequent three sections introduce the three simulation types: discrete-event simulation in Section 1.4, continuous simulation in Section 1.5, and Monte Carlo simulation in Section 1.6. Finally, a basic framework of simulation experimentation is presented in Section 1.7.

1.2 WHAT IS A SYSTEM?

1.2.1 Definitions of Systems

Systems are encountered everywhere in the world. While those systems differ in their specifics, they share common characteristics that lead to a conceptual definition of a system. In Wu [1992], a *system* is defined as "a collection of components which are interrelated in an organized way and work together towards the accomplishment of certain logical and purposeful end." Thus, any portion of the real world may be defined as a system if it has the following characteristics: (1) it has a purpose or purposes, (2) its components are connected in an organized manner, and (3) they work together to achieve common objectives. A system consisting of people is often called a *team*. Needless to say, a mere crowd of people sharing no common objectives is not a team.

When defining a system, the concept of state variable plays a key role. A *state variable* is a particular measurable property of an object or system. Examples of state variables are the number of jobs in a buffer, status of a machine, temperature of an oven, etc. A system in which the state variables change instantaneously at discrete points in time is called a *discrete-event system*, whereas a system in which state variables change continuously over time is called a *continuous system*.

1.2.2 Three Types of Systems

Our universe, which is full of systems everywhere, may be viewed from the five levels of detail (Fig. 1.1): from the subatomic level to cosmological level. In the subatomic level, interactions among the components of a system are described using quantum mechanics, which is a physical science dealing with the behavior of matter and energy on the scale of atoms and subatomic particles. It is interesting to find that quantum mechanics is also used in modeling a system at the cosmological level [Mostafazadeh 2004]. Thus, a system in the subatomic level or cosmological level may be called a *quantum system*.

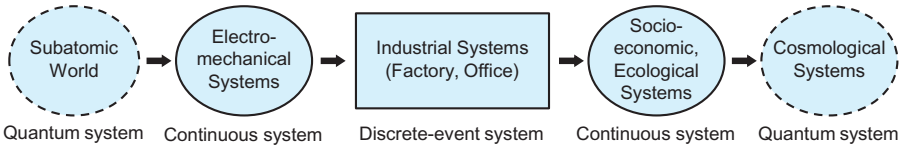


Fig. 1.1. Five levels of details of system definitions in the universe.

A system in the electromechanical level usually has components whose physical dynamics are described using differential equations of effort, such as force and voltage, and flow, such as velocity and current [Karnopp et al. 2000]. The behaviors of ecological systems and socioeconomic systems are usually described using differential equations of flow [Hannon and Ruth 2001]. As a result, these systems are called a *continuous system* or a *differential equation system*.

Systems in the middle level are industrial systems which are more conveniently described in terms of discrete events, and they are discrete-event systems. An event is an instance of changes in state variables. A special type of this system is a digital system such as a computer whose states are defined by a finite number of 0s and 1s.

1.2.3 System Boundaries and Hierarchical Structure

Everything in our world is connected to everything else in some way, which is known as the small world phenomenon [Kleinberg 2000]. Thus, in order to define a system, it is first necessary to isolate the components of the system from the remaining world and to enclose them within a system boundary.

A set of isolated components of primary interest is called a *target system*. The target system may have a number of subsystems, and it may be a subsystem of a higher-level system called a *wider system*. The wider system is separated from the external environment by a boundary [Wu 1992]. In summary, a typical system consists of a target system (composed of its subsystems) and a wider system (in which the target system is included). The system of interest consisting of a target system and its wider system is often referred to as a *source system*.

Most dynamic systems in engineering and management are feedback control systems. Key subsystems in a feedback control system are operational, monitoring, and decision-making subsystems. The operational subsystem carries out the system's tasks, and the monitoring subsystem monitors system performances and reports to the decision-making subsystem. The decision-making subsystem is responsible for making decisions and taking corrective actions. The relationships among the target feedback control system, its subsystems, wider system, and external environment are shown in Fig. 1.2 [Wu 1992]. For example, if your simulation study is focused on an emergency room of a hospital, the emergency room would become the target system and the hospital the wider system.

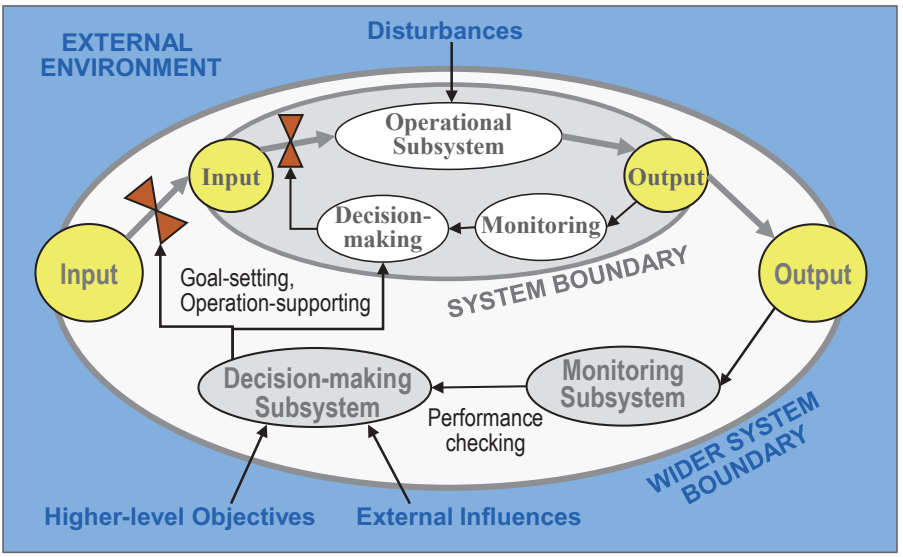


Fig. 1.2. Hierarchical structure of feedback control system.

The wider system influences the target system by setting goals, supporting operations, and checking performances. The target system is subject to disturbances from the external environment. In addition, the external environment provides the wider system with higher-level objectives and other external influences.

Exercise 1.1. Give an example of a feedback control system involving people and identify all the components of the system.

1.3 WHAT IS COMPUTER SIMULATION?

1.3.1 What Is Simulation?

A dictionary definition of *simulation* is "the technique of imitating the behavior of some situation by means of an analogous situation or apparatus to gain information more conveniently or to train (or entertain) personnel." "Some situation" in the definition corresponds to a source system, and an *apparatus* is a simulator. As elaborated in the definition, there are two types of simulation objectives: one is to gain information and the other is to train or entertain personnel. The former is often called an *analytic simulation* and the latter a *virtual environment simulation* [Fujimoto 2000].

The main purpose of an analytic simulation is the quantitative analysis of the source system based on "exact" data. Thus, the simulation should be executed in an as-fast-as-possible manner and be able to precisely reproduce the event sequence of the source system. An analytic simulation is often referred

Fig. 1.3. Examples of virtual environment simulation.

to as a *time-stamp simulation*. A virtual environment simulation is executed in a scaled real-time while creating virtual environments, and it is often referred to as a *time-delay simulation*. Shown in Fig. 1.3 are scenes from a war-game simulation and from a computer game.

An analytic simulation with human interaction is called a *constructive simulation*, and one without human interaction an *autonomous simulation.* If humans interact with the simulation as a participant, it is referred to as human-in-the-loop (HIL) simulation; if machines or software agents interact with the simulation, it is called a machine-in-the-loop (MIL) simulation. A virtual environment simulation without HIL/MIL is often called a *virtual simulation*; one with HIL only a *constructive simulation*; one with both HIL and MIL a *live simulation*. Figure 1.4 shows the classification of computer simulation.

1.3.2 Why Simulate?

Modeling and simulation is the central part of our thinking process. When the situation is too complex to be analyzed by mental simulation alone, we use a computer for simulating the situation. Let's consider the following situations:

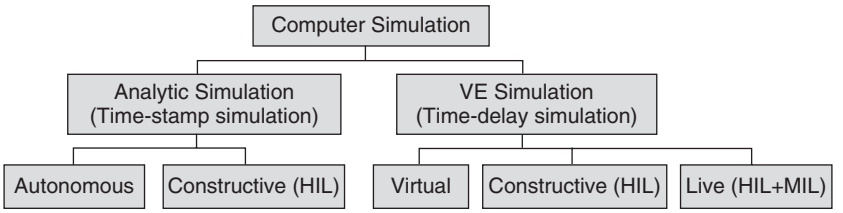


Fig. 1.4. Classification of computer simulation.

1. Finding optimal dispatching rules at a modern 300-mm semiconductor Fab
2. Evaluating alternative designs for hospitals, post offices, call centers, etc.
3. Designing the material handling system of a 3 billion dollar thin film transistor–liquid crystal display (TFT-LCD) Fab
4. Planning a wireless network for a telecommunication company
5. Evaluating high-tech weapons systems for a simulation-based acquisition
6. Designing or upgrading the urban traffic system of a big city
7. Evaluating anti-pollution policies to control pollutions in river systems
8. Evaluating risks in project schedules and financial derivatives

For the above real-life situations, simulation may be the only means to tackle the problems. In practice, simulation may be needed because experimenting with the real-life system is not feasible; your budget does not allow you to acquire an expensive prototype; a real test is risky; your customer wants it "yesterday"; your team wants to test several solutions and to compare them; you would like to keep a way to reproduce its performances later.

The simulation of a discrete-event system is called a *discrete-event simulation*, and that of a continuous system a *continuous simulation*. A class of computational schemes that rely on repeated random sampling to compute their results is referred to as *Monte Carlo simulation*. Among the above situations, Situations 1–6 are concerned with a discrete-event simulation. Situation 7 is concerned with a continuous simulation and Situation 8 with a Monte Carlo simulation.

1.3.3 Types of Computer Simulation

As depicted earlier in Fig. 1.1, the dynamic systems in the universe can be classified into five levels and three types. The three types of dynamic systems are: (1) discrete-event systems, (2) continuous systems, and (3) quantum systems. Thus, it is conceivable that there is one type of computer simulation for each system type. Discrete-event simulation and continuous simulation are widely performed on computers, but the direct simulation of quantum systems